Highwinds CDN Content Protection Products

August 2009

Table of Contents

CDN SECURITY INTRO 3

CONTENT PROTECTION BY CDN DELIVERY PRODUCT 3

HTTP REFERRER 4

ENABLING HTTP REFERRER IN STRIKETRACKER 5

URL SIGNING 6

ENABLING URL SIGNING IN STRIKETRACKER 7
GENERATING A SIGNED PUBLISHING URL 8
VALIDATING A SIGNED PUBLISHING URL 9
PHP CODE 10

GEO BLOCKING 11

RTMPE STREAMING 12

SWF VERIFICATION 13

LIVE STREAMING IP LOCK & LOGIN (PUSH INGEST) 14

HTTP AUTHENTICATION 15

CDN Security Intro

Monetization strategies require content owners to protect their assets from viral distribution. Highwinds gives content providers the ability to create delivery business rules enforced by the CDN. With Highwinds content protection products, end users must view the media through the workflow designated by the publisher.

Content protection policies for many of the Highwinds products are configured inside the StrikeTracker® console. This means the configurations that build restrictions on which end-user requests are honored by the CDN can be independently managed. This guide describes the different security products and shows how to enable them step-by-step.

Content Protection by CDN Delivery Product

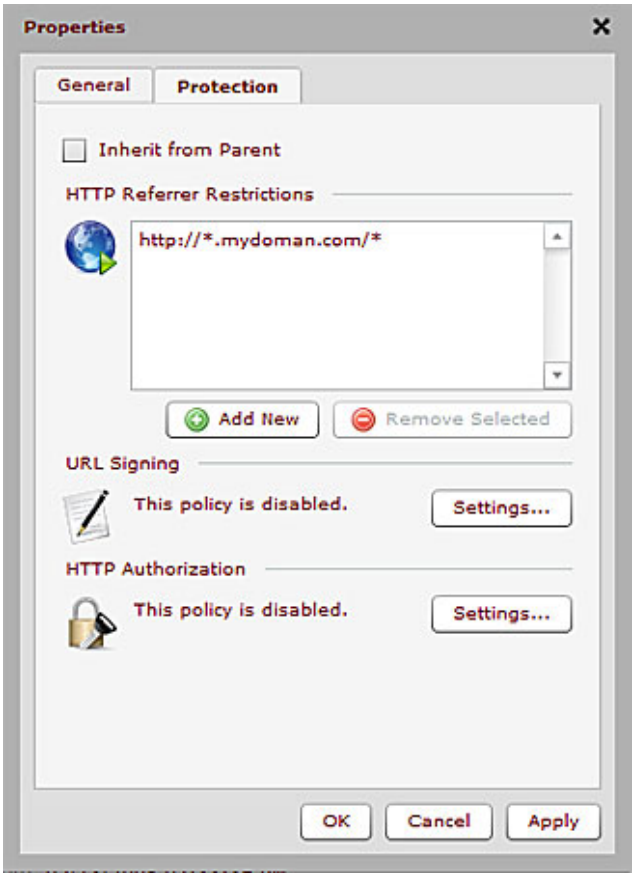
	FMS	WMS	WLS	FLS	CDS
HTTP Referrer	☑	☑	☑	☑	☑
URL Signing	☑	☑	☑	☑	☑
Geo Blocking	☑	☑	☑	☑	☑
RTMPe	☑			☑	
SWF Verification	☑			☑	☑
Live Source Login (Push Ingest)				☑	
Live Source IP Lock (Push Ingest)			☑	☑	
Basic HTTP Auth					☑

If you have any questions about content protection, please contact the Highwinds 24/7 CDN Network Operations Center at CDN-Support@Highwinds.com

HTTP Referrer

HTTP Referrer restriction is a security product that prevents CDN publishing URLs from being freely distributed on unauthorized websites (also known as hot linking or deep linking). Highwinds CDN account owners configure one or more websites that end users can visit and successfully request content hosted by the CDN. When an end user request is made, Highwinds compares the HTTP Header Referrer field with the list of approved websites. If the end user is not visiting from an approved website, the CDN will issue an HTTP 403 – Access Denied response.

Setting up HTTP Referrer security is simple. Policies are enabled on a per-directory basis from within the Content Management tab in StrikeTracker. Follow the steps below to configure and manage these profiles in StrikeTracker.



Enabling HTTP Referrer in StrikeTracker

1. Log into the StrikeTracker account where the desired media is hosted and navigate to the **Content Management** tab.

2. Create or find the subdirectory where the profile needs to be enabled. When enabled on a directory, all files and directories under that tree are included in the profile.

3. **Select** or highlight the target directory in the main navigation window. If all content within a product line should be under the Referrer policy, choose the CDS, FMS, WMS directory (be sure to select this directory in the main viewing window). If a subset of content within a product line should be under the Referrer policy, select the highest directory applicable.

4. Click the **Properties** button in the top navigation bar. Click on the **Protection** tab. Uncheck the box labeled "**Inherit from Parent**".

5. Click **Add New** under the HTTP Referrer Restrictions area of the dialogue box. A pop- up will appear where the allowed domain name needs to be entered. *Syntax is important, since all unaccounted for domains are rejected.* Use wildcards to accommodate sub-domains and URL paths.
    a. Allow all URLs from website: **http://www.mydomain.com/***
    b. Allow all sub-domains URLs on website: **http://*.mydomain.com/***
    c. Special consideration is needed for some versions of some browsers. Not all browsers populate the HTTP Header Referrer field in an expected way. Some browsers omit this field or leave it null. In order to reduce false positives (legitimate end users who are rejected), also allow null HTTP Referrer. Currently, addition of null referrer domain requires a ticket to the Highwinds NOC.

6. Click **OK** to apply the Referrer restriction immediately. Add as many authorized domains as desired. Remove domains by selecting the desired domain and selecting **Remove Selected**.

7. Click **Apply** to exit the Properties dialogue box. The directory with the content protection policy enabled will now have a small golden padlock displayed.

URL Signing

URL Signing is the most popular content protection product offered by Highwinds. Highwinds CDN Account owners use this product to publish content with a query string parameter token that includes a URL expiration timestamp. This private token is created on-the-fly in a server-side implementation, and can be used to create unique publishing URLs for each end user request.

URL security prevents free distribution of content outside the workflow designated by the publisher:

- If an end user tampers with the URL, their request for CDN content is denied.
- If a well formatted URL has an expiration timestamp in the past, end users request for CDN content is denied.

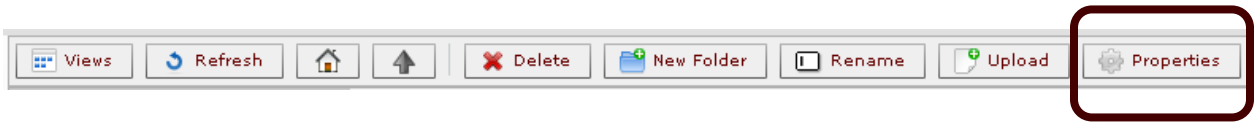
It's easy to take advantage of the Highwinds URL Signing product. First, the URL Signing profile is enabled and managed in the Content Management tab of StrikeTracker. Then with a few lines of web application code, publishers build a URL that's safe from social sharing or deep linking.

URL Signing profiles include the following configuration parameters:

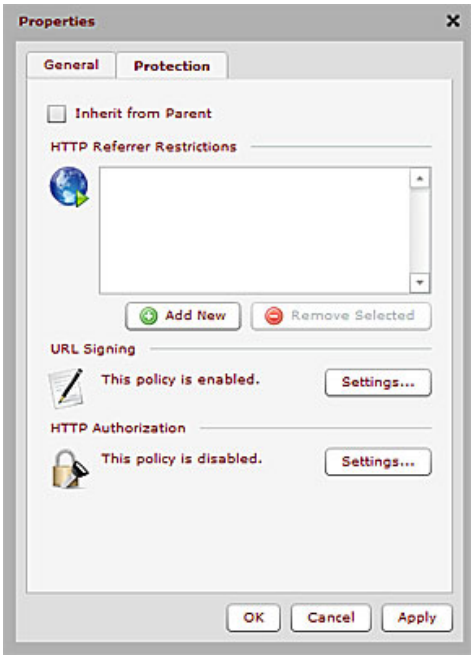
Attribute Name	Description
Pass Phrase Field	URL shared secret parameter **name**, published inside the MD5 hash.
Pass Phrase	URL shared secret parameter **value**, published inside the MD5 hash.
Expiration Field	URL expiration parameter **name**, published in the final URL and also MD5 hashed inside the final token. This is the name of the query string parameter that's published in the final URL. Note that the **value** for the expiration time is generated on-the-fly and is a traditional epoch UNIX timestamp (integer of seconds since midnight January, 01 1970).
Authorized Field	URL token parameter **name**. This is the query string parameter name that's published in the final requesting URL. Name this something unsuspicious (i.e. userPrefs)!
CDN Service Directory	CDN product and optional sub directory to attach this policy to. The policy may be attached to an entire product line for an account, or customers may choose to attach the policy to a sub directory they create. Attaching the policy to a sub directory allows customers to have both secured and unsecured content.

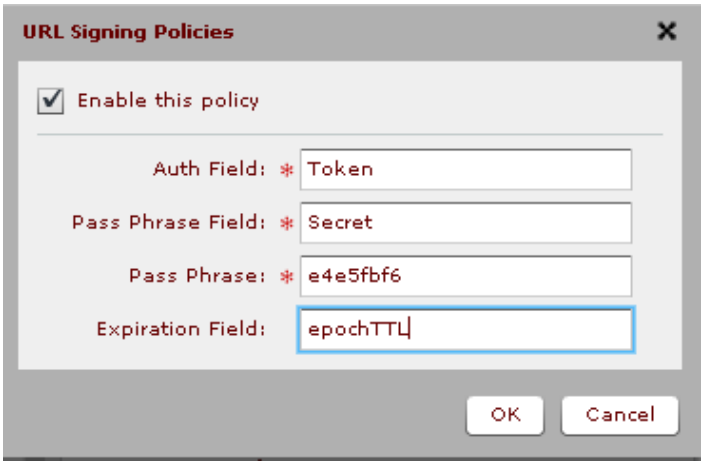
Enabling URL Signing In StrikeTracker

Publishers need to configure a content protection policy on the desired directory. Begin by logging into StrikeTracker and going into the **Content Management** section. Once there, navigate to the product directory or the target folder for secure content. Select the folder in the main navigation window and click the **Properties** button in the title bar. A properties dialogue box is displayed.



In the dialogue box, select the **Protection** tab. Uncheck the box to **Inherit from Parent** and click on **URL Signing Settings**. Enter the desired profile settings. Click **OK** and then click **Apply**.





In the content management directory window, the selected directory is decorated with a golden padlock immediately to show that the real-time Highwinds configuration change is applied. This profile can be modified at any time. Accounts may also have different Content Protection policies for as many different directories or products as desired.

Generating a Signed Publishing URL

1. **Set a URL Signing profile** on the desired directory in the Content Management area of StrikeTracker. For this example, the following profile is setup on the CDN directory listed:
   Auth field: Token
   Pass Phrase field: Secret
   Pass Phrase: e4e5fbf6
   Expiration field: epochTTL
   CDN Directory: /t6a2q6y9/cds/secure/

2. Generate a Time To Live **Epoch Unix timestamp** that is sufficiently in the future for testing the feature. If a time stamp in the past is used then all requests fail. In production, these timestamps are generated in the server-side application code on-the-fly. For this example the following timestamp is used:
   Epoch Unix timestamp: 1437961059
   Human time: Mon, 27 Jul 2015 01:37:39 GMT

3. Start with the Highwinds publishing URL for a file within the directory with the profile.
   http://hwcdn.net/t6a2q6y9/cds/secure/HighwindsDemo.flv

4. Prepare the portion of the URL that will **generate the token**. Remove the http://hwcdn.net and add the query string parameters (name value pairs) for expiration and pass phrase to get the following:
   /t6a2q6y9/cds/secure/HighwindsDemo.flv?epochTTL=1437961059&Secret=e4e5fbf6

   Note: if additional internal query string parameters are used, add them first before adding the URL signing values. Order of these parameters is important.

5. Calculate the MD5 signature of the result of step 4. MD5 libraries are included within most server-side programming languages. MD5 hash generators can also be found online for any manual testing. Note that the secure token output by the MD5 generator is case sensitive. Be sure the MD5 hash generator is not producing an all CAPS token.

   MD5(/t6a2q6y9/cds/secure/HighwindsDemo.flv?epochTTL=1437961059&Secret=e4e5fbf6
   Resulting string: ea6fb765b7b71e50bac2bd5ea9e0ce26

6. Go back to the original Highwinds publishing URL and add the query string parameters (name value pairs) for expiration and the auth token to get the following **secured publishing URL**:

   http://hwcdn.net/t6a2q6y9/cds/secure/HighwindsDemo.flv?
   epochTTL=1437961059&Token=ea6fb765b7b71e50bac2bd5ea9e0ce26

   As in #4 above, order of these parameters is important. First add the expiration name value pair, and then add the token name value pair.

Validating a Signed Publishing URL

1. Start with the secured publishing URL:
   http://hwcdn.net/t6a2q6y9/cds/secure/HighwindsDemo.flv?epochTTL=1437961059&Token=ea6fb765b7b71e50bac2bd5ea9e0ce26

2. Double check the values in the URL Signing profile. Log into the StrikeTracker console, navigate to the Content Management tab and the directory with the golden padlock. Select the directory and the Properties button to view the Protection policies.
   Auth field: Token
   Pass Phrase field: Secret
   Pass Phrase: e4e5fbf6
   Expiration field: epochTTL
   CDN Directory: /t6a2q6y9/cds/secure/

3. Check that the expiration time is not in the past. Online epoch time converters will confirm.
   Epoch timestamp: 1437961059
   Human time: Mon, 27 Jul 2015 01:37:39 GMT

4. Check that the secure token is valid for the URL Signing profile that is configured.

   MD5(/t6a2q6y9/cds/secure/HighwindsDemo.flv?epochTTL=1437961059&Secret=e4e5fbf6)
   Resulting string: ea6fb765b7b71e50bac2bd5ea9e0ce26

5. Keep in mind that:
   - The token is case sensitive. Tokens that are all capital letters will not pass the Highwinds signature check.
   - The order of the query string parameters in the MD5 hashed string and in the final publishing URL matters. First add internal query parameters, then add the expiration URL Signing parameters, and then add the Auth parameters. See #4 and #6 on Generating a Signed Publishing URL.

PHP Code

```
<?php
// Pre-defined values Can be set in StrikeTracker
$usPassPhraseFld = "secret"; // URL shared secret parameter key for input
$usPassPhrase = "user defined"; // URL shared secret parameter value for input
$usExpFld = "expires"; // URL expiration parameter key for input and output
$usAuthFld = "token"; // URL signature parameter key for output

// Signature production code
// File variable will have to be defined dynamically
$domain = "http://hwcdn.net";
$file = "/accoundID/cds/secured folder/filename.example";
$expireTime = time() + (30); //30 seconds expiration

//Steps 1-4 in generating a link for URL signing.
$signing_url = $file . "?" . $usExpFld . "=" . $expireTime . "&" .
$usPassPhraseFld
. "=" . $usPassPhrase;

//MD5 Function called in PHP Step 5
$signature = MD5($signing_url);

//Step 6
$output_url = $domain . $file . "?" . $usExpFld . "=" . $expireTime . "&" .
$usAuthFld . "=" . $signature;

//Outputing URL to Screen for example
print $output_url;
?>
```

Code Output

Signature hash input
/t6a2q6y9/cds/secure/HighwindsDemo.flv?epochTTL=1437961059&Secret=e4e5fbf6

Signature hash output
ea6fb765b7b71e50bac2bd5ea9e0ce26

Final URL
http://hwcdn.net/t6a2q6y9/cds/secure/HighwindsDemo.flv?epochTTL=1437961059&Token=ea6fb765b7b71e50bac2bd5ea9e0ce26

GEO Blocking

Highwinds GEO Blocking allows publishers to restrict content to end users in specified locations. The IP address of incoming requests is checked against a current list of IP allocations to Countries and States within the US. If an end user's IP address is not found in the list, they are allowed access to the content by default. The feature has both an Include and an Exclude list which are used to target the allowed audience.

Geo Blocking Granularity: Country, US State, US City, US Zip Code, DMA

GEO Blocking is not yet in the StrikeTracker™ portal and is currently enabled only through a Highwinds NOC support ticket. To request a GEO Block profile, send an email to cdn-support@highwinds.com. Include Highwinds Account ID, target directory for this content protection profile and a list of Country codes or State codes to include or exclude. Please also send the NOC a sample URL to a file in the specified directory.

Example:

Attention Support:	Please enable a GEO Block policy
Account ID:	a2a3a4a5
Product Line:	CDS
Folder:	USOnly
Include:	US
Exclude:	ALL but US
Test Link:	http://hwcdn.net/a2a3a4a5/cds/USOnly/myfile.wmv

Implementation Best Practice

GeoBlocking on Live Flash or Live Windows Media is enabled on a per Account ID basis. Once enabled, the feature applies to all streams within the CDN account. If multiple GeoBlock profiles are desired or if both secure and unsecure streams are desired, segment out the streams in CDN sub accounts.

RTMPe Streaming

RTMPe is fast, real-time encryption supported by the Flash Media Server that secures data transfer between the server and the client. This feature prevents third-party applications from listening to, and perhaps "ripping" the stream. RTMPe is enabled on a per-request basis and is available for both Flash On-Demand and Flash Live.

The RTMPe feature is requested by appending this following Highwinds query string parameter to the publishing URL:

dopproto=rtmpe

Request the following Flash On-Demand publishing URL and Highwinds returns a playlist containing RTMPe edge URLs:

http://hwcdn.net/z3m6y2h2/fms/NYSubwayReef.flv.xml?dopproto=rtmpe

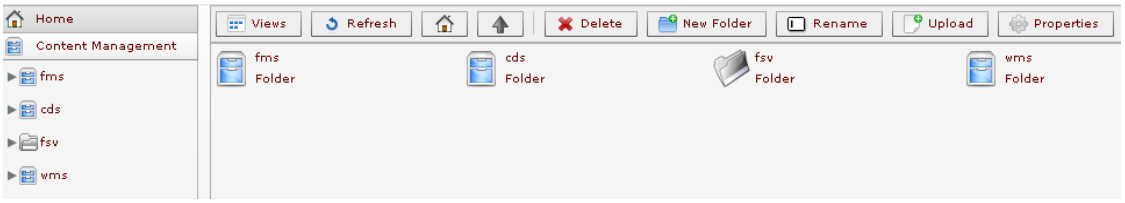
Implementation Best Practice

RTMPe streaming is enforced with URL Signing. When combined with URL Signing, end users will only be able to access content via RTMPe. If URL Signing is not used, the end user can access rtmp urls by simply removing the query string parameters dopproto from the publishing URL. Details on enabling and implementing URL Signing are in this document.

SWF Verification

SWF Verification is an Adobe Flash Media Server feature that compares the SWF playing in the client with one or more SWFs approved by the content publisher. Highwinds FMS servers inspect both the Flash player size and the Flash player hash, or the last 32 bytes of the first handshake packet. If the players are not an exact match, the end user is blocked from viewing the stream. This feature prevents manipulated or foreign players from accessing the video.

SWF Verification is a popular content protection product on Highwinds. No code changes in the player are needed to support SWF Verification. This product is enabled on a per-account basis, meaning that all Flash video live or on-demand within the account needs to be delivered to an approved player.



The steps to enabling the feature are:

1. Request the feature once by sending the NOC a support request. Email cdn-support@highwinds.com and include the Account ID to enable the feature.

2. Log into the FTP space for the account and upload all approved SWF files into the new **fsv** directory shown beside the product directories (FMS/CDS/WMS). FTP must be used to upload the SWFs, though the **fsv** directory will appear in the StrikeTracker Content Management area and the FTP space.

3. End users must view the content through one of the approved players. Be sure any player updates are uploaded to the fsv directory before being published live.

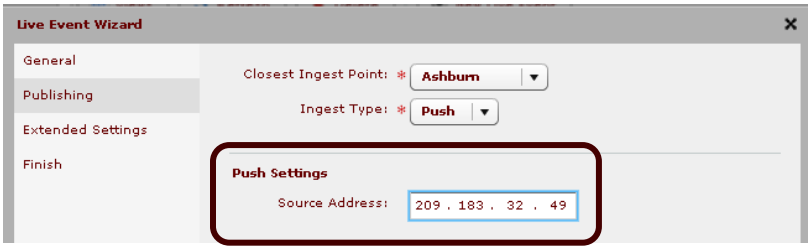
Additional information about SWF Verification is available on the Adobe website:

http://livedocs.adobe.com/flashmediaserver/3.0/docs/help.html?content=03_configtasks_22.html

Live Streaming IP Lock & Login (Push Ingest)

Highwinds provides two methods of preventing stream source hijacking on Live Push ingest.

IP Lock allows only a specified IP address to provide the source stream to a Highwinds push publishing point. This product is supported for both Windows Live Push and Flash Live Push, where Push is the method of getting Highwinds a seed or source feed for the live video stream. This feature is enabled per-stream in the StrikeTracker live stream provisioning wizard:



Login requires an authentication step for an encoder that wants to push a seed stream to a Highwinds Live Flash publishing point. This feature is enabled per-stream, is currently supported for Flash Live only, and starts with a support ticket.

- Request the feature once per stream by sending the NOC a support request. Email cdn-support@highwinds.com and include the Account ID, the stream publishing URL, and the desired username and password.

Note that a ticket is needed to enable login for Live ***Push*** Ingest, but not needed to enable login for Live Pull Ingest. If the live stream source requires Highwinds to authenticate before accessing the ingest or seed stream, this is configured in the StrikeTracker live stream provisioning wizard where the source address is specified.

HTTP Authentication

Highwinds supports Basic HTTP Authentication for delivery on the CDS product line. With Basic HTTP Authentication, end users are prompted to enter Login credentials that are approved by the customer's web server before the media is delivered. HTTP Authentication policies are enabled and managed in the StrikeTracker portal.

Basic HTTP Authentication profiles include the following required fields:

- **Binding Point**. This is the URL location for secured authorization. This URL is a secured file, page or directory where Highwinds will make an HTTP HEAD request to validate the user credentials it receives. The Binding Point must be an HTTP URL; SSL is not supported at this time. When configuring a Web server to serve as the auth binding point, it's important to make sure that the server will require authentication for HEAD requests, not just GET and POST.

  Example binding point: http://www.mydomain.com/secure/index.html

  In this example index.html will have security configured so that a user name and password file is used for validation.

  For information on how to create basic authentication on your web server, please see the provided link for Apache. If you are using another server type your user manual should provide the same information.

  http://httpd.apache.org/docs/1.3/howto/auth.html

- **Connect Count**. This is a maximum number of connections Highwinds will allow at once to the auth binding point. This is an integer value, and applicable **per instance** (2 per facility). This parameter is configurable in order to throttle request load on the customer's Web server. To keep end user experience prompt during peak times, set this number high.

- **TTL**. This is the number of seconds that Highwinds caches a successfully authenticated user's session. When an end user is successfully authenticated, Highwinds asks the user agent to set a cookie containing an encrypted authentication token, and this token expires in **TTL** seconds. Effectively, a given user should only be authenticated against the configured binding point once every **TTL** seconds. For best results, this value should be just above the user's average time on the site. If a user is spending an average of 15 minutes on the site you might want the TTL to be 1080 for 18 minutes.

- **Realm**. This is the name of the authentication realm given back to the user on requests which do not contain auth credentials. For HTTP Basic Auth, this value is usually displayed by the browser to the user when login credentials are requested. Set this to something familiar, so the end user understands the source of the request.

As with the existing content protection methods, basic auth can be configured on a per-directory basis. To setup the HTTP Basic authentication, go into the **Properties** of the sub folder and select **Protection**.